Comparison of Available Gaming Technologies

Chuck McLean
Program Manager - Simulation and Visualization
National Institute of Standards and Technology
Gaithersburg, MD

Supporting Naval Education and Training Command
Pensacola, FL

Topics

- ***Needs Statement***
- ***Game Engine Comparison Criteria***
  - ***Graphics and sound***
  - ***Game engine management***
  - ***Character animation***
  - ***Physics***
  - ***Artificial intelligence***
  - ***Game editor and development tools***
  - ***Application data import and export***
  - ***Multi-player operations and server support***
  - ***Software distribution and security mechanisms***
  - ***Learning system support***
- ***Questions and Discussion***

Needs Statement

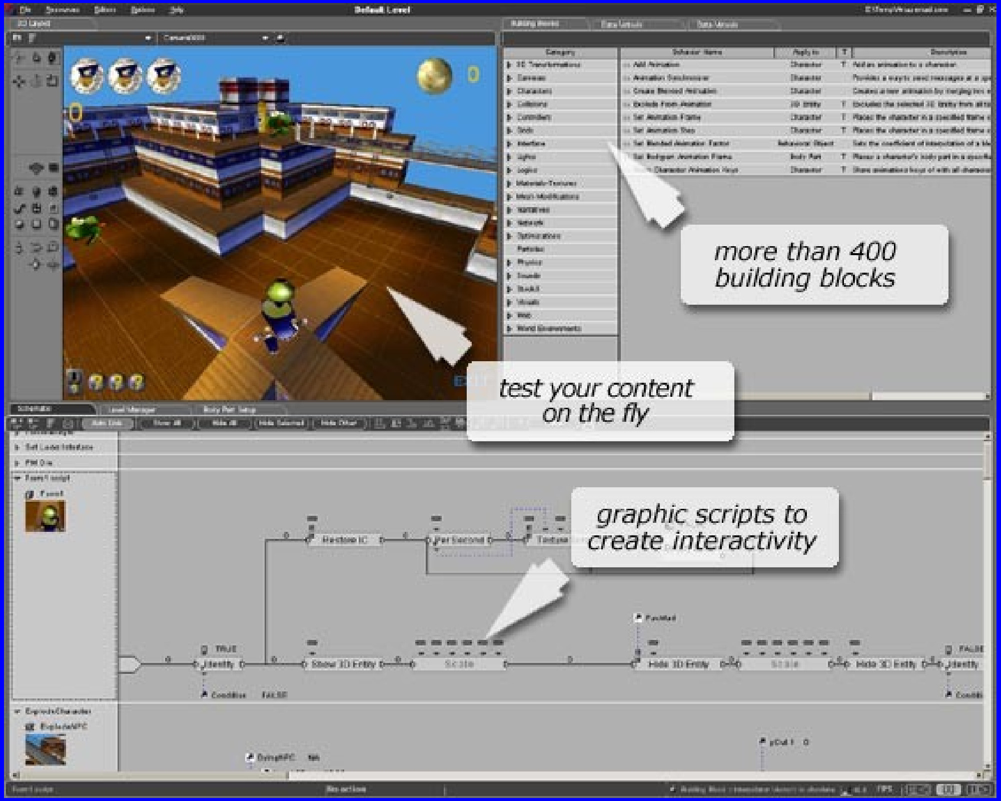
- ***Navy needs a game software engine to support the development and delivery of PC simulation-based learning content***
- ***Required capabilities will include both traditional course management system, game engine, and simulator functions***
- ***Software engine must interface with existing and future Navy Learning Management Systems and Learning Content Management Systems***
- ***Software engine must support selected Navy PC platform and peripheral device configurations***
- ***Must support security policies and access mechanisms***
- ***Forward and backward compatibility for legacy systems and interfaces must be supported over long Navy system life cycles (I.e., decades)***

Graphics and Sound

- ***Generate images from two and three-dimensional objects/scenes***
- ***Map textures onto objects***
- ***Provide various forms of lighting, cast shadows***
- ***Model fog and smoke***
- ***Provide multiple cameras (viewing controls)***
- ***Graphics object libraries and procedurally-generated graphical objects***
- ***Multiple sound files, ambient sound, varying locations, voice generation from text***

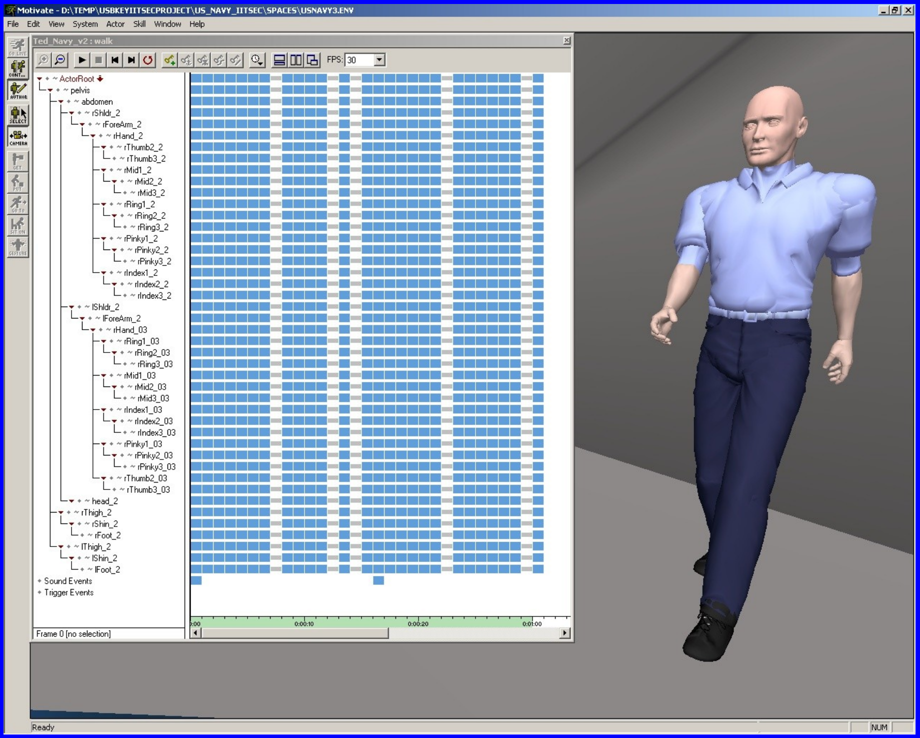
Game Engine Management

- ***Core of the system***
- ***Implements game genres, e.g., role-playing, strategy, first person shooter, etc.***
- ***Execution control logic, supervisory capabilities, coordinates other modules***
- ***User interface management, display of multiple windows***
- ***Scripting capabilities***
- ***System clock and calendar, management of time, triggers, scheduling of events.***
- ***Saving and loading game levels and game state, pausing execution, playback and roll back***


more than 400 building blocks
test your content on the fly
graphic scripts to create interactivity

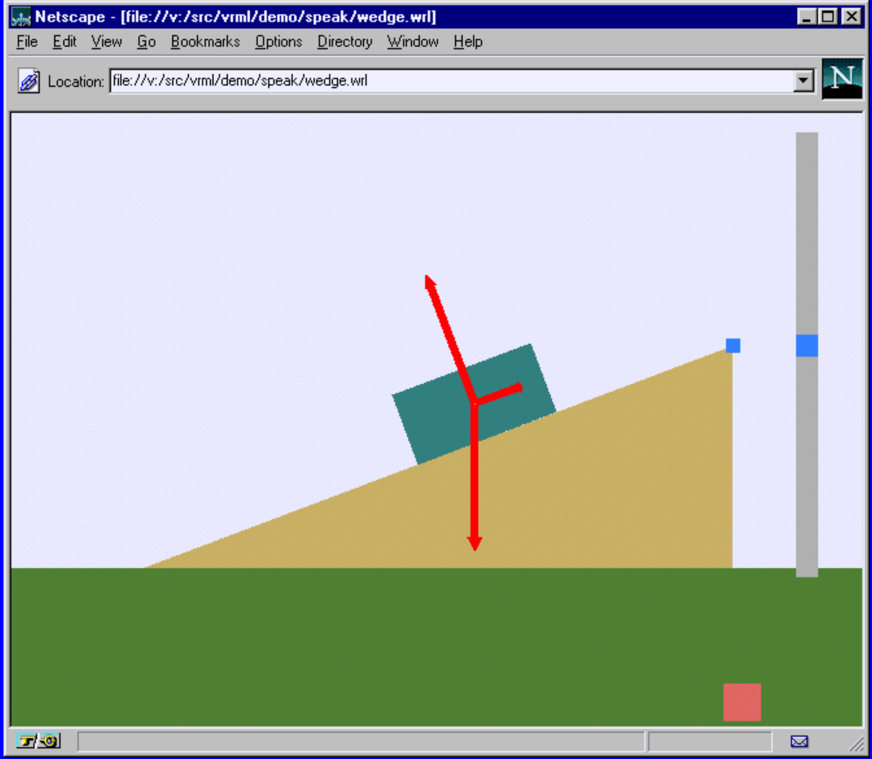
Character Animation

- ***Development of character models, bones (controls for body segments), range of motion for body joints***
- ***Key frame animation***
- ***Implementation of intelligent behavior, footstep locomotion***
- ***Smooth interpolation of body joint regions when characters are in motion and blending of different motion sequences***
- ***Motion capture***
- ***Cloth and hair, physics of character interaction, manipulation of props and objects***



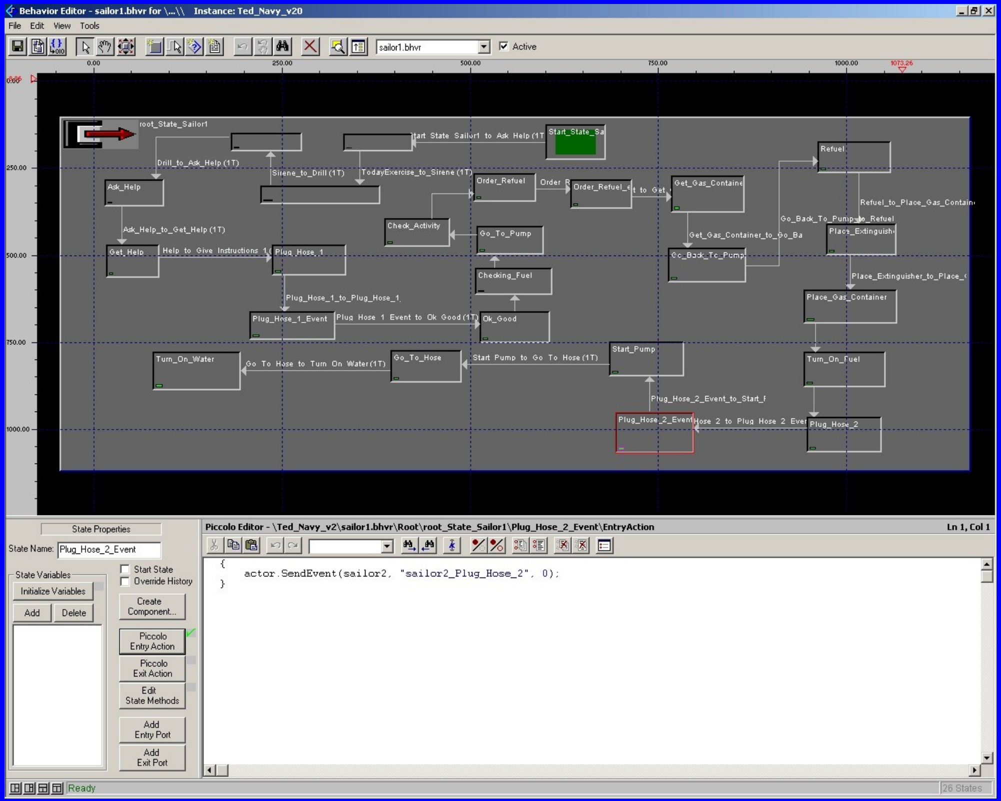
Physics

- ***Performs calculations to enable realistic display of collisions, trajectories of projectiles, falling objects, cornering of vehicles, floating and sinking objects (e.g., vessels), movements of cloth and hair, etc.***
- ***Soft or rigid body mechanics, proximity sensors are required to implement interactions between objects.***
- ***Data: mass, center of mass, volume, Newton's laws, inertia, units of measure, geometry and vectors, kinematics of objects, linkages, accelerations, velocities, forces and torques, momentum***


Netscape - [file://v:/src/vrml/demo/speak/wedge.wrl]
File Edit View Go Bookmarks Options Directory Window Help
Location: file://v:/src/vrml/demo/speak/wedge.wrl

Artificial Intelligence

- ***Various techniques for path planning, strategies of non-player characters, coordinated or goal-directed behaviors, flocking, smart terrain, automatic solution of lower level behaviors for player characters, character motion, learning, speech recognition and natural language processing, randomness and statistical behaviors***
- ***Genetic algorithms, neural networks, fuzzy logic***
- ***Rule-based systems, decision trees, problem solving systems***



Behavior Editor - sailor1.bhvr for \...\\   Instance: Ted_Navy_v20

Piccolo Editor - \Ted_Navy_v2\sailor1.bhvr\Root\root_State_Sailor1\Plug_Hose_2_Event\EntryAction   Ln 1, Col 1

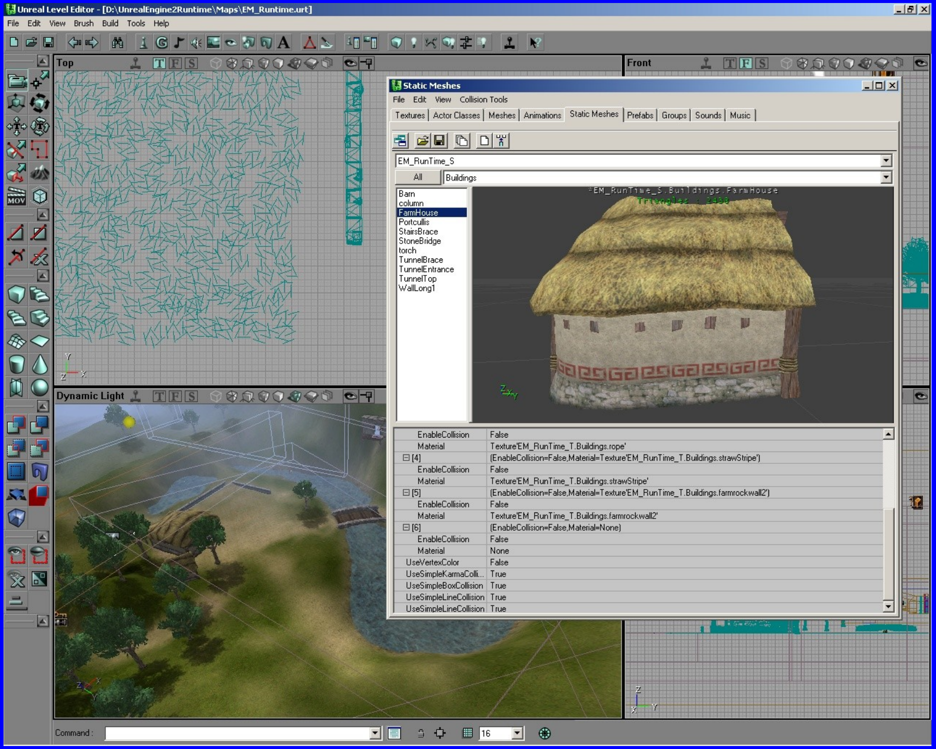
State Name: Plug_Hose_2_Event

☐ Start State
☐ Override History

```
{
    actor.SendEvent(sailor2, "sailor2_Plug_Hose_2", 0);
}
```

Game Editor and Development Tools

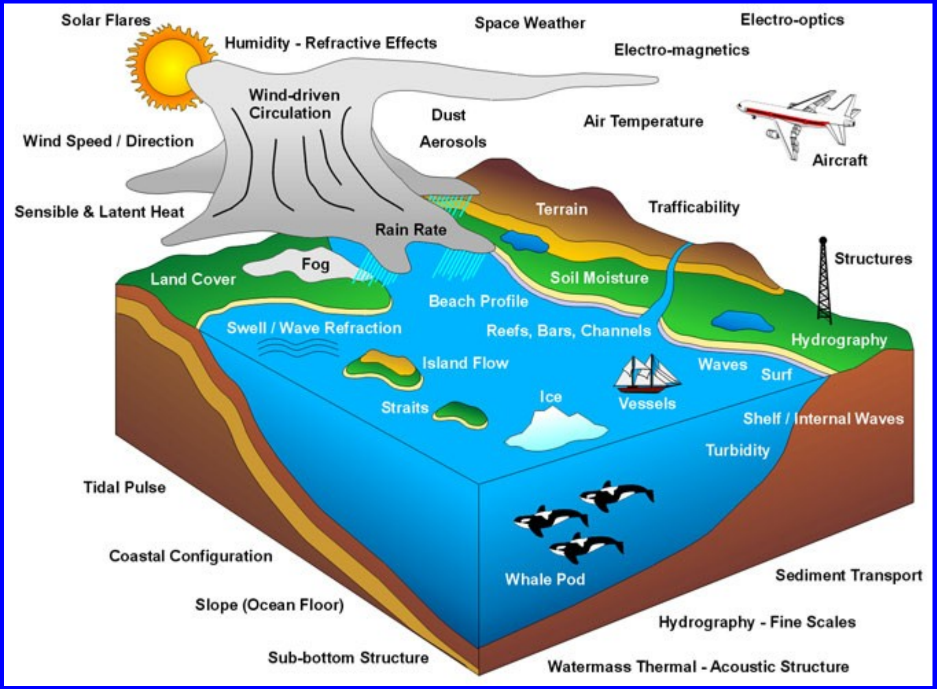
- ***Primary way of developing content - allows work at a high level to create environments, player and non-player characters, and supporting objects***
- ***Create behaviors on characters and objects, define game levels, establish lesson plans, scoring schemes, etc.***
- ***Provides a scripting language, previewing and debugging capabilities for testing scripts, and a seamless interface to the game engine itself.***
- ***Well-documented and easy to use with on-line help & a rich set of examples***


Unreal Level Editor - [D:\UnrealEngine2Runtime\Maps\EM_Runtime.urt]
File Edit View Brush Build Tools Help
Top
Front
Dynamic Light
Static Meshes
File Edit View Collision Tools
Textures | Actor Classes | Meshes | Animations | Static Meshes | Prefabs | Groups | Sounds | Music
EM_RunTime_S
All
Buildings
Barn
column
FarmHouse
Portcullis
StairsBrace
StoneBridge
torch
TunnelBrace
TunnelEntrance
TunnelTop
WallLong1
Command :
16

EnableCollision	False
Material	Texture'EM_RunTime_T.Buildings.rope'
⊟ [4]	(EnableCollision=False,Material=Texture'EM_RunTime_T.Buildings.strawStripe')
EnableCollision	False
Material	Texture'EM_RunTime_T.Buildings.strawStripe'
⊟ [5]	(EnableCollision=False,Material=Texture'EM_RunTime_T.Buildings.farmrockwall2')
EnableCollision	False
Material	Texture'EM_RunTime_T.Buildings.farmrockwall2'
⊟ [6]	(EnableCollision=False,Material=None)
EnableCollision	False
Material	None
UseVertexColor	False
UseSimpleKarmaColli...	True
UseSimpleBoxCollision	True
UseSimpleLineCollision	True
UseSimpleLineCollision	True

Application Data Import and Export

- *Support import and export of data in as many appropriate standard and proprietary data formats as possible*
- *Relevant formats include graphics files, video, audio, game levels, game state, character models, motion capture, RLOs, XML objects, etc.*
- *Learning content based upon Reusable Learning Objects (RLOs) that are SCORM-compliant using Extensible Markup Language (XML).*
- *Example - SEDRIS - standard for representing environmental objects using XML*

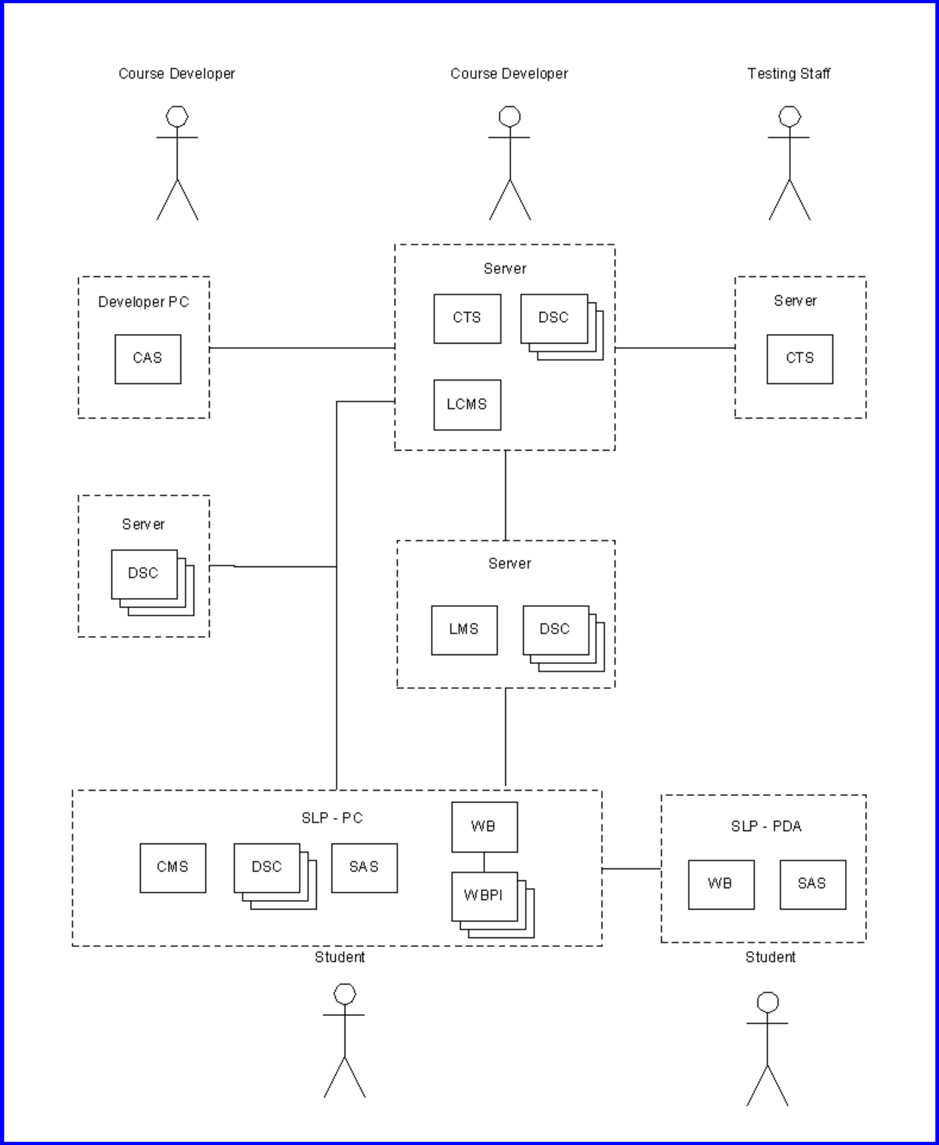


Multi-player Operations and Server Support

- ***Support team-training environment for classrooms and ad hoc teams***
- ***Employs distributed simulation mechanisms such as High Level Architecture (HLA) Run-Time Infrastructure (RTI) or Massively Multi-Player (MMP) games***
- ***Voice Over Internet Protocol (VoIP), a mechanism for carrying out voice communications between players and instructors***
- ***Issues with how and when players receive information on fellow players actions***

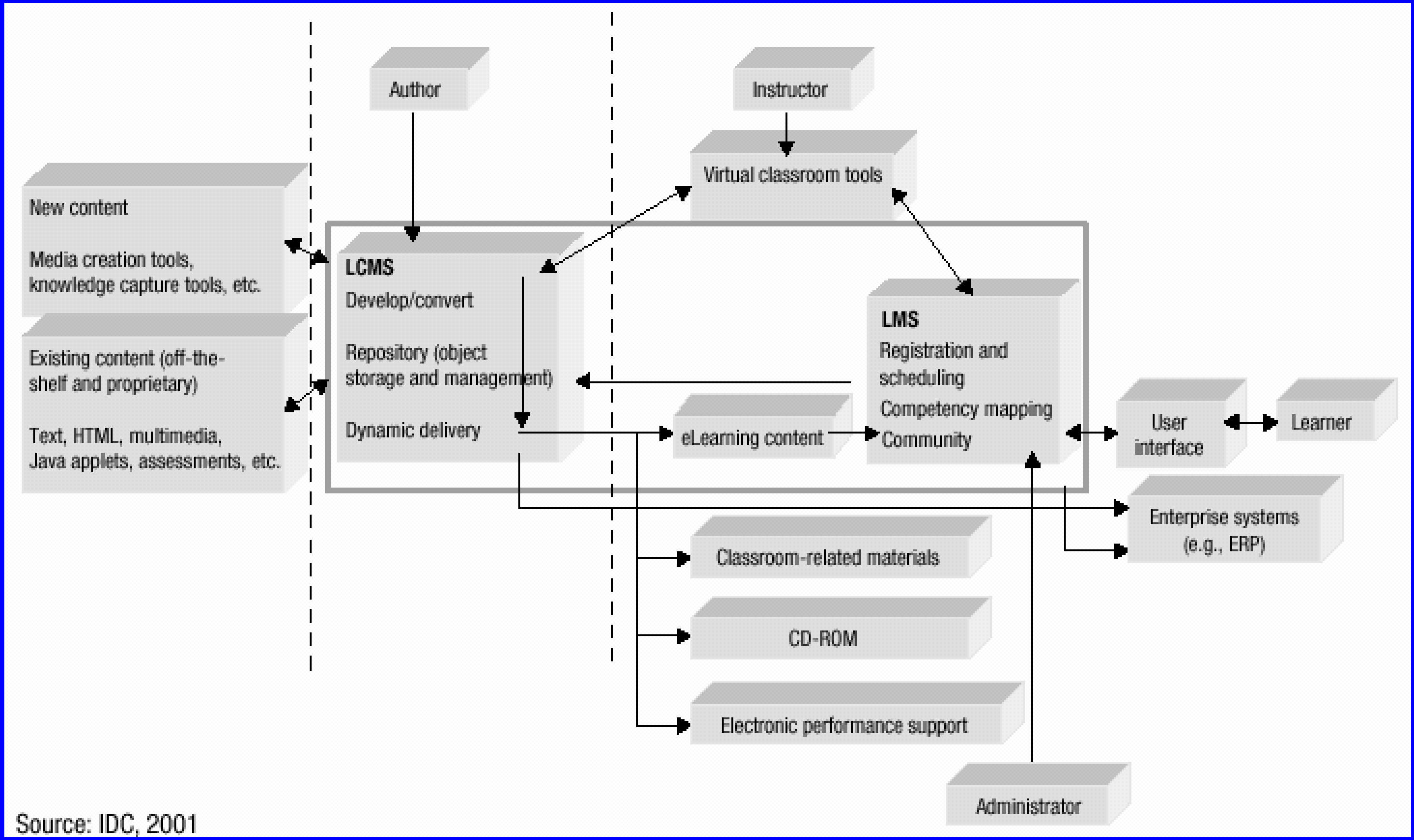
Software Distribution and Security Mechanisms

- ***Distribute the video game-based learning application and courseware over the Internet as Web page downloads.***
- ***Avoid use of CD-ROMs or physical media of any sort, long download times during course execution, large software footprints on user machines are undesirable.***
- ***Security constraints associated with the Navy Marine Corps Intranet (NMCI) environment must also be addressed.***



Learning System Support

- *Video game engines need to be integrated with Learning Content Management Systems (LCMSs) and Learning Management Systems (LMSs) to deliver course content to the student, manage instruction, vary lesson scenarios provide on-line help, and track the student’s progress*



Author

Instructor

Virtual classroom tools

New content

Media creation tools, knowledge capture tools, etc.

Existing content (off-the-shelf and proprietary)

Text, HTML, multimedia, Java applets, assessments, etc.

LCMS

Develop/convert

Repository (object storage and management)

Dynamic delivery

LMS

Registration and scheduling

Competency mapping

Community

eLearning content

User interface

Learner

Enterprise systems (e.g., ERP)

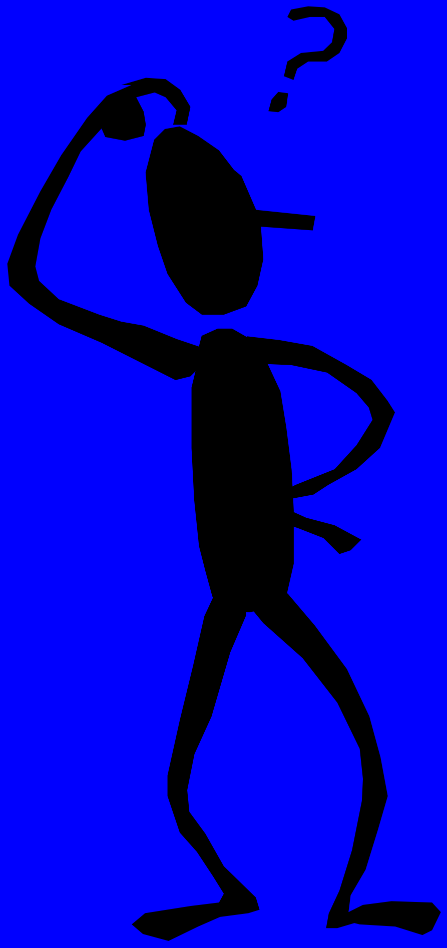
Classroom-related materials

CD-ROM

Electronic performance support

Administrator

Source: IDC, 2001



Questions and discussion?

For further information, please contact:

Chuck McLean (301) 975 -3511

Email: charles.mclean@nist.gov

Program Manager - Simulation and Visualization

Building 220 Room A127
National Institute of Standards and Technology
Gaithersburg, MD 20899